

RESEARCH

Open Access



Smart container monitoring using custom-made WSN technology: from business case to prototype

Peter Ruckebusch^{*} , Jeroen Hoebeke, Eli De Poorter and Ingrid Moerman

Abstract

This paper reports on the development of a prototype solution for tracking and monitoring shipping containers. Deploying wireless sensor networks (WSNs) in an operational environment remains a challenging task. We strongly believe that standardized methodologies and tools could enhance future WSN deployments and enable rapid prototype development. Therefore, we choose to use a step-by-step approach where each step gives us more insight in the problem at hand while shielding some of the complexity of the final solution. We observed that environment emulation is of the utmost importance, especially for harsh wireless conditions inside a container stacking. This lead us to extend our test lab with wireless link emulation capabilities. It is also essential to assess feasibility of concepts and design choices after every stage during prototype development. This enabled us to create innovative WSN solutions, including a multi-MAC framework and a robust gateway selection algorithm.

Keywords: Internet-of-things, Wireless sensor networks, Remote monitoring and configuration, Container monitoring

1 Introduction

Over the last few years, several research aspects of wireless sensor networks (WSNs) have been explored. Most publications investigate in depth a particular protocol, algorithm, or architecture. Rarely, they consider an integrated solution that needs to be deployed in a real-life setting and that covers all layers of the protocol stack, including the actual application. Creating a prototype that needs to operate in such a setting, in particular outdoor, still remains a challenge, especially when including solutions tailored to a specific use case.

In this paper, we present an approach to develop such a prototype. This methodology has been applied to the design of a battery-powered tracking device for a smart container monitoring solution that combines a GPS locator, a GPRS modem, and a IEEE-802.15.4 transceiver. The research and development started from a clear business case with specific requirements in mind for the WSN software, focusing heavily on cost reduction and energy efficiency. A major requirement was the ability of

the tracking devices to create an inter-container network using WSN technology in order to reduce roaming costs, minimize data connections, and reduce overall energy consumption. Since the tracking device has to work in an environment that is hostile for wireless communication, robustness is a key factor. From the beginning, it was clear this was a non-trivial task that could not be completed by directly applying readily available software.

During 2 years, both academic and industrial partners, combining expertise in different domains, have successfully cooperated in order to build a prototype that was capable of fulfilling both business and technical requirements [1]. The resulting prototype will be used as the base for an industry grade product. In this paper, we reflect on the applied methodology, illustrating it with concrete results, and lessons learned. We hope this work can inspire others developing prototypes for challenging environments, filling the gap between research and product development.

The remainder of this introduction introduces the container monitoring business case before an outline of the rest of the paper is given.

There is a clear need for end-to-end container and cargo monitoring. Nowadays, containers often get lost during

^{*}Correspondence: peter.ruckebusch@ugent.be
Department of Information Technology (INTEC), Ghent University - imec - IDLab, Technologiepark-Zwijinaarde 15, 9052 Ghent, Belgium

transport, and tracking them is a time-consuming task. Because no information about container location and itinerary is available, a lot of empty containers are transported around the globe. With more detailed information, this can be avoided and the number of containers on the road can be drastically reduced. Moreover, if the cargo is monitored, problems are detected earlier and accountability is easier. If tracking and monitoring of containers is combined with an intrusion detection system, customs can assign a higher priority to those containers, reducing delay during transit. This is in line with the increasing interest and need for end-to-end supply chain visibility. It can be assumed that smart tracking devices will play a crucial role in the green trade lanes of the future and will fundamentally change the world of container transport.

However, in order to turn this into reality, the proposed solution must be accepted in a conservative sector where profit margins are very low. As such, operational expenses should be minimized. These include the replacement of batteries and the communication costs. Moreover, the process of replacing batteries must avoid reverse logistics and should be included in the regular container maintenance, scheduled every 1 to 3 years. The initial investments should also be reasonable, allowing for a gradual rollout, as discussed in [2]. The installation and maintenance of the devices should not involve skilled labor, and since there are many users and stakeholders, interaction with the device must be kept as simple as possible without neglecting security concerns. A more detailed discussion about the business case can be found on [3].

Because of the high business potential, several solutions for container monitoring have been proposed already [4–9]. Most of them only use a combination of a GPS locator with a GPRS or satellite modem. Some of them also include a RFID, IEEE802.11, or IEEE802.15.4. There are however several differences with the solution presented in this paper, among others:

1. The tracking devices create a low-power inter-container network and share one GPRS/GPS connection in order to reduce overall energy consumption, avoid roaming costs, and use less data connections.
2. The software running on the tracking devices is configurable in terms of application and network parameters. This allows for a more fine grained control.
3. Specifically tailored sensor communication solutions are used in order to improve network performance and to extend battery lifetime.

Figure 1 gives an overview of the different steps that were required to go from business case to prototype. The outcome of each step is also summarized, given an overview of the contributions. The paper is structured

accordingly. First, it was necessary to check if communication is feasible in a container environment (Section 3) before designing the system architecture and the tailored tracking device software (Section 4). Then, we verified if the envisioned system could meet the business requirements mainly in terms of device battery lifetime (Section 5). Next, the solution was thoroughly evaluated in a test lab environment (Section 6). Finally, a prototype tracking device was developed and tested in a field trial (Section 7).

2 Related work

Since this paper reports on the approach rather than on an extensive evaluation of a solution, we will also focus our related work discussion on this particular aspect of real-life deployments. Other container monitoring solutions, discussed in the introduction, do not describe a formal method. Therefore, we will consider other deployment scenario's.

The importance of having a systematic approach was already clearly stated in reports about WSN deployments that failed or did not deliver the expected results. Most of them lacked a good initial feasibility study and a thorough evaluation on a test lab before going into the field. For instance, Langendoen et al. [10] describe the lessons learned after a 1-year pilot WSN deployment in a potato field. They foremost stress the importance of initial test lab evaluations.

Cerioti et al. [11] reported on a WSN-based system for adaptive, closed-loop control of lighting in road tunnels. They followed a similar road map as us, first investigating the wireless characteristics in tunnels [12] before designing the system. Then, they created a test setup in the field. This defers from our method because we did not have a container stacking permanently available. Moreover, due to the mobile nature of containers, this would only cover a subset of the possible scenario's of a tracking device.

Barrenetxea et al. [13] report about their experiences with SensorScope [14], an environmental monitoring WSN system in several deployments. They offer a guide to the reader for deploying an outdoor WSN. Beside numerous tips and tricks, they also discussed the different stages of a field trial: development, testing, and deployment. They also noted that emulation is a crucial aspect during the testing stage. Although implicitly mentioned, they do not include feasibility studies (i.e., characterizing wireless communication and calculating expected energy consumption) in their road map. In our case, this was a necessary step before we could start the design and development stage.

3 Container communication feasibility study

Creating a communication network between containers was put forward as the key requirement from the

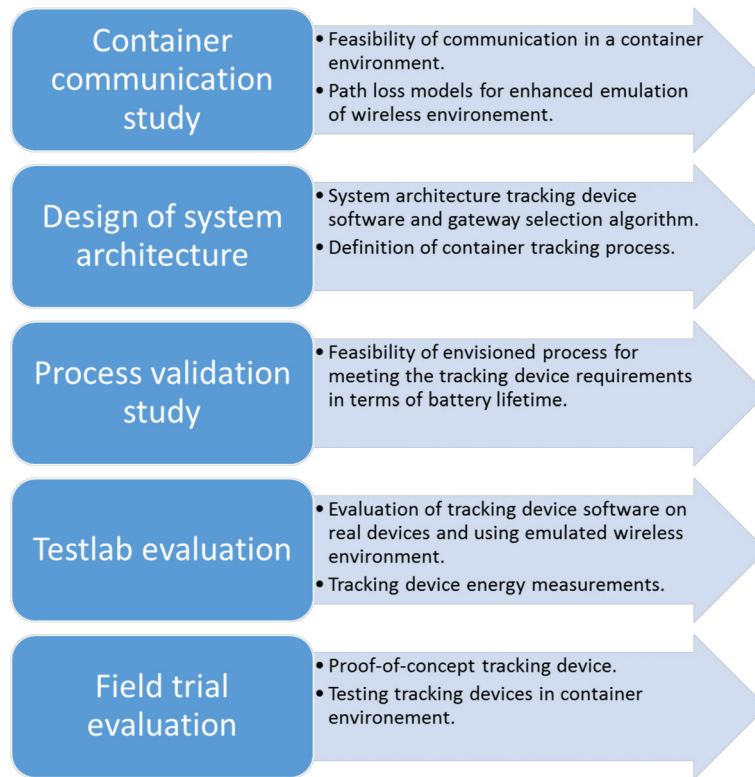


Fig. 1 Overview of the different steps to go from business case to a successful prototype implementation. The outcome of each step is also mentioned

beginning of the project. Because stacked metal containers are highly reflective, wireless communication suffers from multipath fading. Therefore, before starting the design, implementation, and evaluation phases, it was crucial to investigate the feasibility of communication with, between, and inside containers. In this section, we will discuss if it is possible to:

- Set up a wireless link between containers and a cellular network, i.e., extra-container communication
- Create an ad hoc network between stacked containers, i.e., inter-container communication
- Communicate from outside to inside the container and vice-versa, i.e., intra-container communication

Path loss was calculated for each of the container communication types (i.e., extra, inter, and intra), allowing to analyze the feasibility. They are discussed in the next subsections. For the extra- and inter-container communication types, also, path loss models were created, which are useful to simulate the wireless environment in tools such as NS-3 and Cooja, allowing more realistic simulations. However, it was important to verify if the path loss models are representative for real hardware by comparing with RSSI measurements in the same environment. The last subsection covers this aspect.

3.1 Measurement setup and data processing

The path loss measurements were conducted at a container repair site in the port of Antwerp. The measurement setup consists of a transmitting and a receiving part. At the transmitter, a signal generator creates a continuous wave which is fed to the transmitting antenna Tx. At the receiving end, a spectrum analyzer samples the power at the receiving antenna Rx. As Tx and Rx, vertically polarized half-wave dipole antennas are used. The transmitter antenna was attached to the side of the container near the ventilation holes, the intended location of the tracking device.

Following measurements, path loss (PL) in decibels is calculated as in Eq. 1:

$$PL = P_{Tx} - L_{Tx} - L_{Rx} - P_{Rx} \quad (1)$$

In Eq. 1, P_{Tx} is the transmit power (20 dBm), L_{Tx} and L_{Rx} are the Tx and Rx antenna feeder losses in dB, and P_{Rx} is the received power in decibel-milliwatts. It is important to note that the usual calculation of path loss also includes terms in the right-hand side of (1) which exclude the gains of the measurement antennas. This was however not done here: due to the proximity of the metallic container surface, antenna effects cannot be separated from



Fig. 2 Measurement environment at a container repair site in the port of Antwerp for extra-container communication. The figure includes a picture of the actual environment (left side) and a schematic top-down view (right side)

the wireless propagation loss, and as such, the path loss in Eq. 1 includes both phenomena.

3.2 Extra-container communication

The extra-container link is implemented with GSM and/or UMTS. Path loss is therefore investigated for the GSM/UMTS frequencies of 900, 1850, and 2100 MHz outside a large stack of 40-ft containers. Figure 2 includes a picture of the container configuration (left side) and a schematic top-down view (right side). The containers are stacked 4 to 7 high. The pathways between container rows are about 1.5 m wide. The Tx is mounted near the ventilation holes at the top of a ground level container. The Rx is attached at the same height as the Tx to a mast on a cart which is pushed along several tracks outside the stack (striped area in Fig. 2) while the spectrum analyzer continuously samples the received power. The extra-container path loss samples are smoothed out using a sliding window of length 3 m to remove small-scale fading. Per frequency, on average, 973 smoothed path loss samples are obtained. These path loss samples cover distances between the Tx and Rx from 10 up to 63 m.

Path loss (PL) (in dB) is found to correlate well with logarithmic distance (frequency average correlation coefficient of 0.86) and is therefore fitted to the model in Eq. 2 as function of distance d (in m) between Tx and Rx:

$$PL(d) = b_0 + b_1 \cdot 10 \log_{10}(d) + \chi_s \quad (2)$$

In (2), b_0 and b_1 are regression parameters (b_1 is the path loss exponent) and χ_s is a zero-mean random variable that accounts for shadow fading. χ_s assumes a normal distribution with standard deviation σ_s . Table 1 lists the estimated parameters of Eq. 2. The path loss exponent b_1 is smaller than 2 for all three frequencies which is probably a consequence of waveguiding effects existent in the stacked container environment.

Based on the above path loss model and using a maximum allowable path loss of 140 dB, the estimated wireless

range for the extra-container communication frequencies are ± 3 km. Because this distance is obtained after extrapolating path loss data collected up to 63 m, it only serves as an indication for the communication range. In the following, we assume that

Assumption 1 *It is possible to use cellular networks inside a stacked container environment for extra-container communication.*

3.3 Inter-container communication

For the inter-container communication, two often occurring container stackings were investigated: (a) row-stacking and (b) block-stacking. Figure 3 illustrates the measurement environment for both row-stacking (left side) and block-stacking (right side). The upper part of the figure gives a schematic top-down view and indicates the position of Tx (squares) and Rx (circles). The measurements are repeated on 433, 868, and 2400 MHz.

3.3.1 Row-stacking

Measurements are performed along a row of four 40-ft containers (total length of 42.7 m). As depicted on Fig. 3, the Tx is placed near the ventilation holes of the first container. The Rx is then moved in steps of 0.5 m at the same height as the Tx along the container row, resulting in 83 Rx locations where the median of 200 samples of received power is recorded. Measured path loss in decibels is found

Table 1 Extra-container path loss model parameters

Freq. [MHz]	b_0 [dB]	b_1 [–]	σ_s [dB]
900	70.19	1.82	3.22
1850	85.34	1.27	3.17
2100	85.88	1.22	3.30

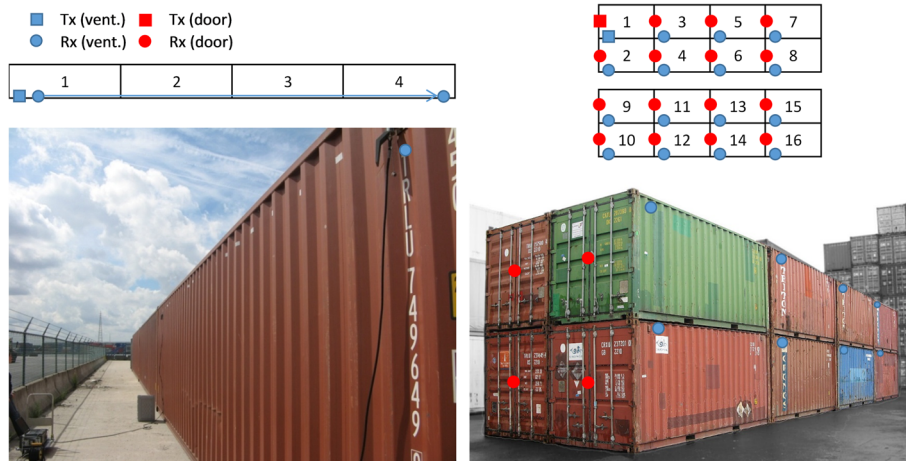


Fig. 3 Measurement environment at a container repair site in the port of Antwerp for inter-container communication. The left side of the figure illustrates a row-stacking and the right side, a block-stacking. For both configurations, a picture of the actual environment (lower part) and a schematic top-down view (upper part) is included

to correlate well with logarithmic distance (average correlation of 0.78 over the three frequencies). Path loss (PL) (in dB) is therefore fitted to the model in Eq. 2.

Table 2 lists estimated values for the parameters of (2), obtained by least squares fitting using our measurement results. The 95% confidence bounds on the vertical location of the regression lines are at most ± 2.80 dB over all frequencies and distances. Taking this uncertainty into account, it can be concluded from Fig. 4 that path loss is nearly identical for all frequencies at large distances. At small distances, path loss is similar for 433 and 868 MHz but is higher for 2400 MHz. This can be attributed to the periodically ribbed structure of container surfaces (period = 27 cm, Fig. 3). At smaller distances, surface waves along the container dominate propagation. In contrast to 433 and 868 MHz, the wavelength at 2400 MHz is smaller than the ribbed structure's period, which means this frequency is more prone to diffraction losses at the rib edges.

3.3.2 Block-stacking

Measurements were performed on a 3D-container stack where 16 20-ft containers are stacked 4 long, 2 wide, and 2 high, as illustrated in Fig. 3. Two locations for Tx and

Rx were considered, i.e., near the ventilation holes (blue) and near the door handles (red). For both positions, path loss is measured between the Tx on container 1 (squares) and the Rx's on the other containers (circles). For each Tx-Rx link, the median of 300 samples of received power is recorded.

The path loss samples are fitted to the regression model in (2). Table 3 lists the estimated regression parameters. The average correlation between path loss and logarithmic distance is 0.76. Figure 5 shows measured path loss versus distance for the three frequencies and both scenarios. The 95% confidence bounds on the vertical location of the six regression lines are ± 4.94 dB on average over all frequencies and distances.

Table 2 Path loss model parameters for a row-stacking container environment

Freq. [MHz]	b_0 [dB]	b_1 [-]	σ_s [dB]
433	47.38	2.24	5.49
868	47.64	2.09	6.56
2400	54.58	1.48	6.19

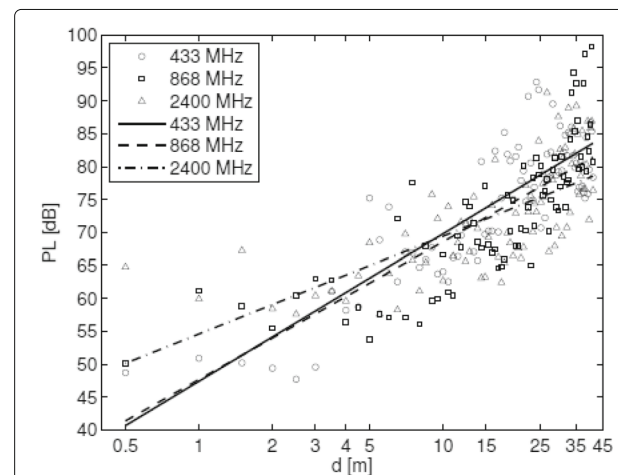


Fig. 4 Path loss vs. distance for inter-container communication in a row-stacking

Table 3 Path loss model parameters for a block-stacking container environment

Freq. [MHz]	Vent-mounted			Door-mounted		
	b_0 [dB]	b_1 [–]	σ_s [dB]	b_0 [dB]	b_1 [–]	σ_s [dB]
433	55.46	2.52	7.76	53.36	1.64	3.98
868	58.10	2.38	7.98	43.93	3.25	4.94
2400	42.69	2.90	9.30	45.50	2.95	8.33

For the vent-mounted scenario, path loss is nearly the same at 433 and 868 MHz and comparatively smaller at 2400 MHz. Propagation between Tx and Rx for this scenario occurs mainly along the small gaps between two containers (widths of around 10 cm), due to the fact that both Tx and Rx are closed in between containers. Propagation in small gaps is less lossy at 2400 MHz due to the smaller wavelength.

For the door-mounted scenario, path loss is similar at all frequencies for smaller distances and relatively smaller at 433 MHz for larger distances. For this scenario, the Tx is not enclosed between containers and the dominant propagation paths also exist partly in the free space around

the container stack. For larger distances, a larger portion of propagation path length is situated in free space, where the lowest frequency of 433 MHz exhibits the least path loss.

The results in Tables 2 and 3 and Figs. 4 and 5 show that inter-container communication is possible, leading to the following assumptions for the remainder of the project:

Assumption 2 *Inter-container communication is possible, but the wireless ranges are limited to one 40-ft container in a block stacking and four 40-ft containers in a row stacking.*

3.4 Intra-container communication

The path loss for intra-container links is investigated for 433, 868, and 2400 MHz. The antenna placement for the intra-container measurements is shown in Fig. 6. The Tx is mounted outside the container on the ventilation holes, and the Rx is installed at two different locations inside the container. The first Rx location (Rx₁) is inside, directly facing the Tx on the outside. The second Rx location (Rx₂) is inside at the container's center.

Intra-container measurements were carried out for both 20- and 40-ft containers. The 20-ft container is an older type without ventilation holes, allowing to analyze the effects of having ventilation holes on the path loss. Both containers were closed and empty during measurements; therefore, additional attenuation by the cargo has to be

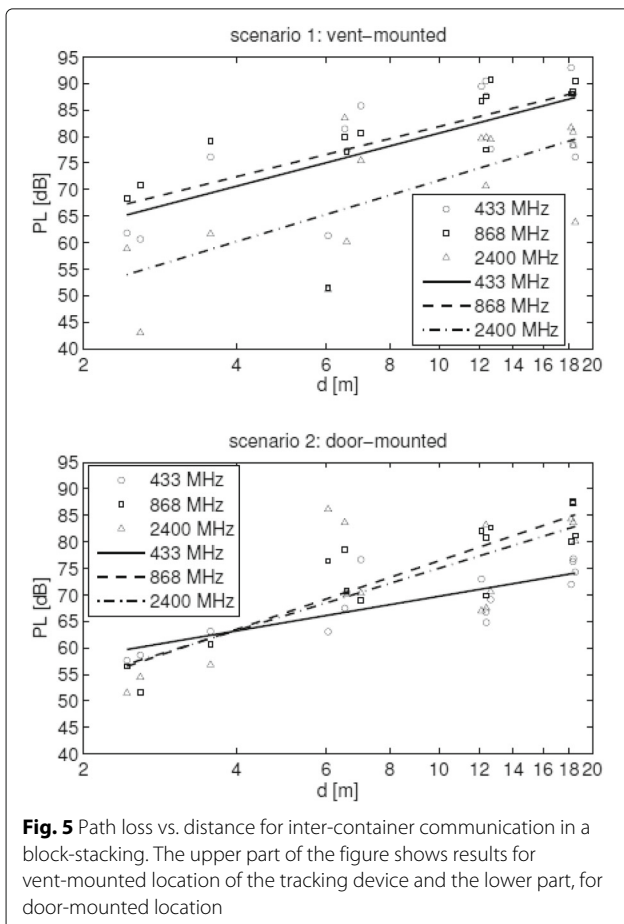


Fig. 5 Path loss vs. distance for inter-container communication in a block-stacking. The upper part of the figure shows results for vent-mounted location of the tracking device and the lower part, for door-mounted location



Fig. 6 Measurement setup for path loss calculations inside a container (intra-container link)

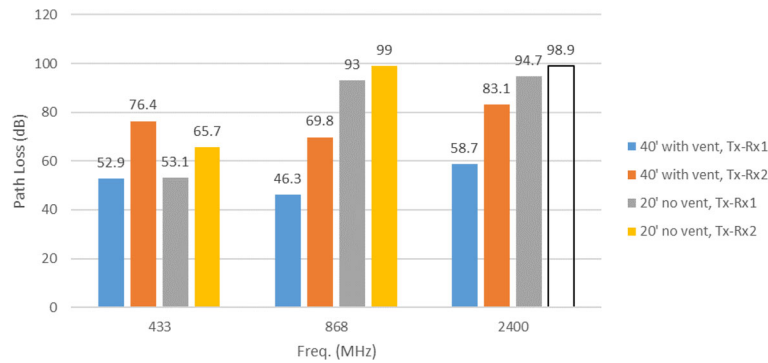


Fig. 7 Path loss (in dB) for intra-container communication in a 20-ft container without and a 40-ft container with ventilation holes. Two Rx locations and three frequencies are considered

added to the path loss. For each of the two Tx-Rx links, 200 samples of the received power P_{RX} are recorded and used to calculate path loss PL using Eq. 1. Figure 7 shows the maximum value of PL over 200 samples for each of the frequencies, container types, and Tx-Rx links. The transparent bar for Tx-Rx₂ inside the 20-ft container at 2400 MHz means that the maximum path loss exceeded the maximum sensitivity of the spectrum analyzer (the maximum measurable path loss is mentioned on top of the bar).

Figure 7 shows that the maximum path loss values for 40-ft containers (with ventilation holes) allow intra-container communication for all three frequencies and for both Tx-Rx links. The path loss for Tx-Rx₂ is about 27 dB larger than for Tx-Rx₁. For the 20-ft container without ventilation holes, it can be concluded that a reliable intra-container link is not possible for 868 and 2400 MHz. A reliable link is however feasible for 433 MHz. This leads to the following assumption:

Assumption 3 *Intra-container communication is possible on 433, 868, and 2400 MHz when a container has ventilation holes. Otherwise, only 433 MHz can be used.*

3.5 Comparing path loss model with RSSI measurements

One of the side benefits of performing path loss measurements is that we are now able to model the wireless environment inside a container stacking and use this model in simulators such as NS-3 and Cooja. However, since the path loss measurements were performed with specialized equipment (signal generator and spectrum analyzer), it was necessary to verify if a path loss model is representative for actual hardware. To this end, RSSI measurements were taken using Sentilla sensor nodes equipped with a CC2420 transceiver. Simultaneously, path loss measurements were conducted with the measurement equipment placed as close as possible to the sensor nodes as depicted

in the right side of Fig. 8. A top-down view of the container stacking is depicted on the left side of Fig. 8.

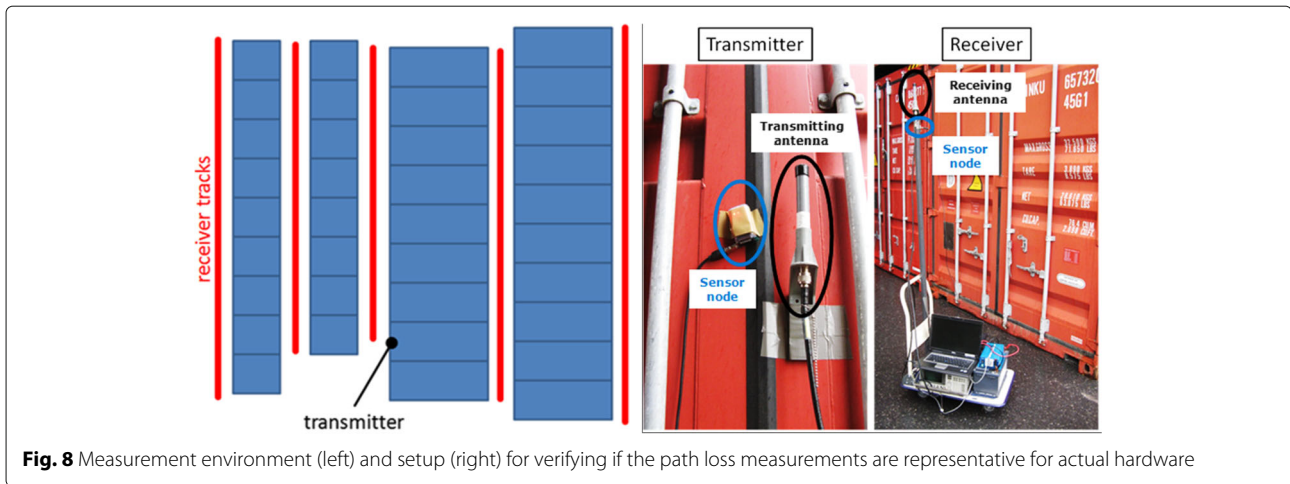
Figure 9 shows measured path loss versus transmitter-receiver distance (on a logarithmic scale) for both the signal power measurements (blue markers) and the RSSI measurements (red markers). The global increase of path loss with distance is determined through linear regression. This is done separately for the signal power and RSSI measurements: see the blue and red straight lines in Fig. 9, respectively.

The main observation in Fig. 9 is that the slope of both straight lines appears to be nearly the same, indicating that both received signal power and RSSI decrease at the same rate with distance. This observation is numerically confirmed by a statistical t test that concludes that there is no significant difference between the slopes of both straight lines (t test p value of 0.44).

Practically, this means that the conversion between received signal power and RSSI is linear: between received signal power and RSSI, there is a constant (i.e., distance- and thus power-independent) shift of 6.0 dB. RSSI is therefore a good measure of received signal power. This also follows from the specifications of CC2420 chip used in the Sentilla sensor nodes: herein, a good RSSI linearity of ± 3 dB is mentioned. This leads to the following assumption:

Assumption 4 *Path loss models can be used as input for wireless link emulation.*

Since lost packets do not contribute to the path loss models, it was also important to verify the packet loss with real hardware. We noticed a packet loss between 0 and 20% for distances that fall in the wireless range calculated in Figs. 4 and 5. When going out of this range, the packet loss quickly rises, exceeding 60% at 18 m. Hence, we can safely assume the following:



Assumption 5 *It is possible to establish inter-container links on a small scale. Large-scale networks will need multihop communication.*

In the remainder of the project, we have chosen to use 2400 MHz. The choice of this frequency has both practical (the sensor nodes in our test lab operate on 2400 MHz) as technical (this ISM band is free to use globally) reasons.

4 Design of the system architecture

After determining that inter-container communication was indeed feasible, the next step was the design of the system architecture. This section gives a high-level overview of the different components (architectural entities), processes, and novel sensor communication solutions.

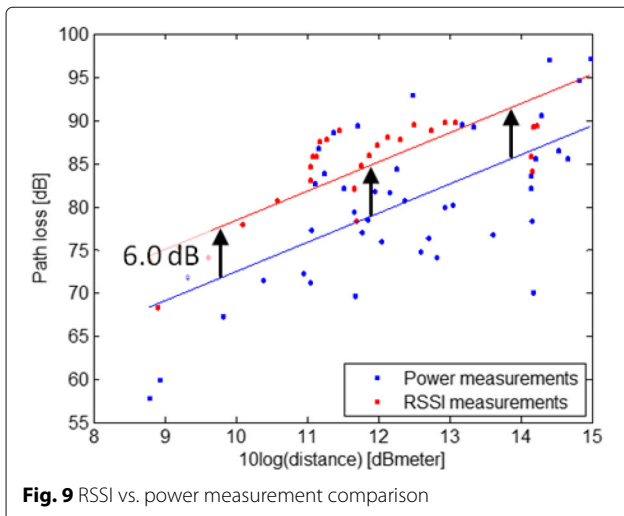
4.1 Component and connection overview

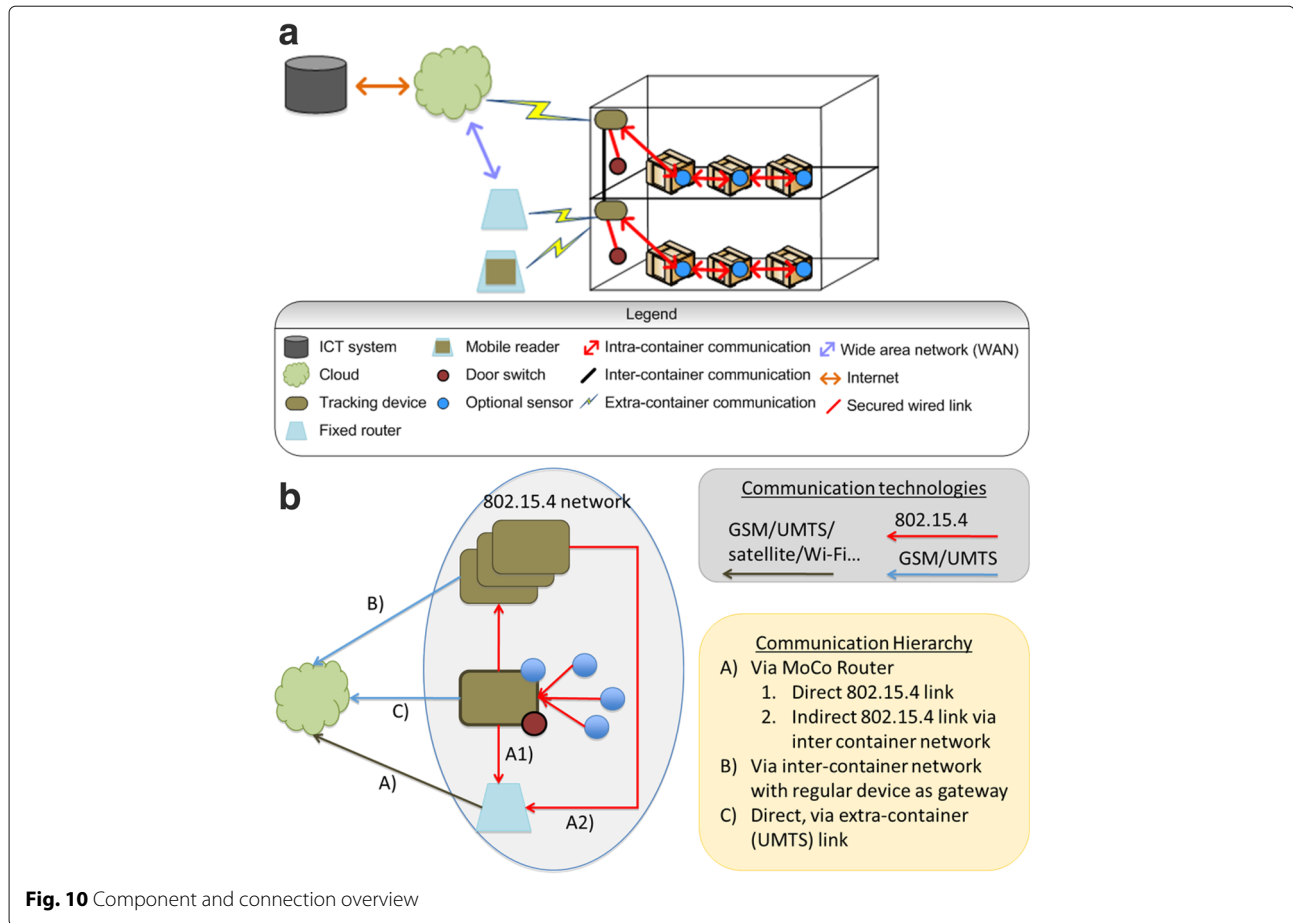
In Fig. 10, a high-level overview of the general architecture is shown. The main components of the system are

depicted in part a of Fig. 10. The tracking device (brown oval) is a lightweight battery-powered device that will be mounted on a container and will become an unbreakable part of that container. In addition, it is connected via a secured wired link to a door sensor (light brown circle) capable of monitoring opening and closing of container doors. The tracking device is equipped with a cellular radio for direct connectivity with the central ICT system, a GPS for obtaining the containers location, and a IEEE-802.15.4 radio for short-range, low-power communication. The latter enables to create an ad hoc network with neighboring tracking devices and with sensors inside the container (blue circle) that monitor the status of the cargo.

Beside the tracking device, there are also two types of special devices: (1) an optional MoCo router (light blue trapeze), installed on key infrastructure for container handling such as cranes, boats, and lorries, and (2) a MoCo reader (light blue trapeze with brown oval inside) integrated in the handhelds (e.g., smartphone) of container operators. Both devices serve a special role. The MoCo router is mains-powered and can be used as gateway for tracking devices. It enables to collect tracking reports and monitoring data in an energy efficient manner (i.e., by avoiding the usage of the cellular radio). The MoCo reader is used by container operators to interface with the tracking device. For instance, when a container is loaded and sealed, a container operator can activate the door sensor, and conversely, when a container is unloaded, an operator can disable the door sensor.

The tracking device will periodically report status information (position, battery status, etc.) to the Cloud, will inform about critical events (e.g., opening of doors after sealing), and, optionally, will send data collected by sensors attached to the cargo consisting of sensitive and high-value products (e.g., temperature, toxic gasses). Listing 1 illustrates the content of periodical reports, comprising





42 bytes in total. It contains the tracking device ID, the container ID, a sequence number, GPS location information, a timestamp, the status of the door sensor (i.e., contact), a msgID, and the state of the tracking device. The state contains the mode of the tracking device (i.e., STP, SVE, or REST) and a substate (i.e., transmitting or not transmitting). The defined states will be explained in detail in Section 4.1.

Listing 1: Content of periodical report.

```
typedef struct gps_data {
    float latitude;
    float longitude;
    float velocity;
    float height;
} gps_data_t;

typedef struct MoCoState {
    uint8_t mode;
    uint8_t subState;
} MoCoState_t;

typedef struct Report {
    uint16_t nodeID;
    uint32_t sequenceNumber;
    uint8_t containerID[12];
    gps_data_t gpsData;
    uint32_t dateTime;
    uint8_t contact;
    uint8_t msgID;
    MoCoState_t state;
} Report_t;
```

Part b of Fig. 10 illustrates the different communication types. The sensors attached to the cargo will communicate with the tracking device over IEEE-802.15.4. This is intra-container communication. In order for a tracking device

to transmit its status to the central ICT system, several possibilities exist. The device can directly send the data by turning on and using its UMTS/GPRS radio or can forward it over IEEE-802.15.4 to a nearby router, thereby saving energy. This is called extra-container communication. Alternatively, in order to save battery, the tracking device can also transmit the information to neighboring tracking devices, sharing UMTS/GPRS connectivity. This is called inter-container communication. Using these different communication capabilities, the architecture is capable of delivering all information required by customers.

One of the key features for saving energy and reducing (communication) cost that has been incorporated in the architecture is the ability to reduce UMTS/GPRS usage by making use of IEEE-802.15.4 communication (in between tracking devices and with a fixed router). The cost aspect is straightforward. For the energy consumption, the underlying motivation is that UMTS/GPRS communication is much more energy-consuming than IEEE-802.15.4. Each reduction in UMTS/GPRS usage contributes significantly to the energy reduction and lifetime of the tracking device. This will be shown in Section 5. Of course,

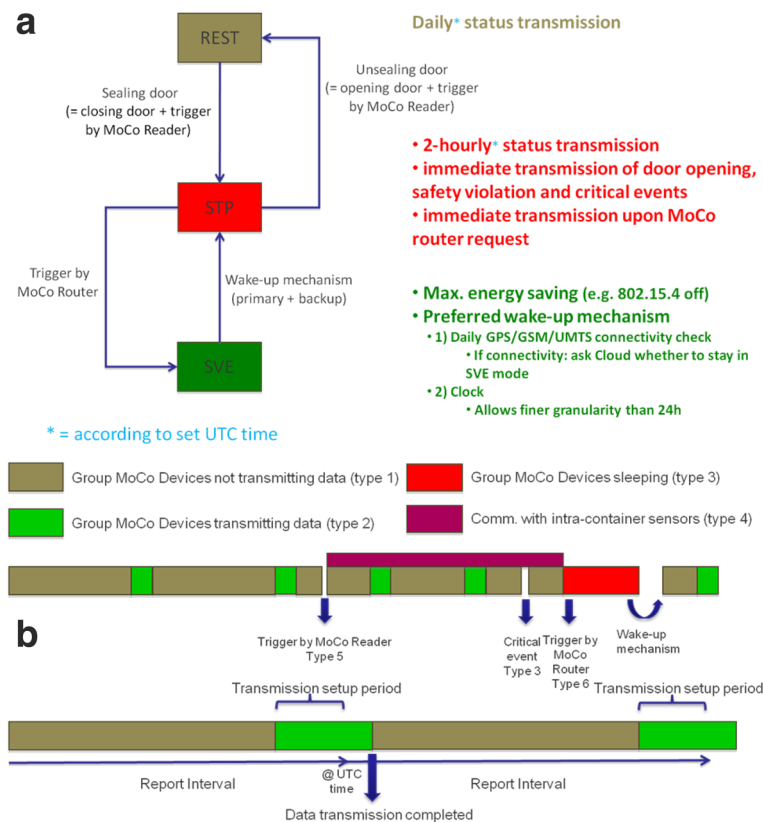


Fig. 11 Communication and process overview

achieving this reduction will depend on the presence of neighboring devices or routers, which will become more significant with an increasing uptake of the solution.

4.2 Process overview

The container monitoring process, illustrated in part a of Fig. 11, was defined taking into account the typical stages found in the logistic transportation chain. This allowed to optimize the communication strategy as much as possible.

When a container is stored empty, its tracking device will daily report about its position and battery level according to a fixed Universal Time Coordinated (UTC) time (e.g., 12:00 UTC time). This way, neighboring containers will all transmit around the same time and communication between containers can be more easily optimized: a single tracking device can turn on its UMTS/GPRS radio and share it with other devices. This status is called REST mode.

When the container is stuffed, its doors are sealed and an operator triggers the tracking device to go in Secured Transport Mode (STP mode). This trigger can be sent using a mobile reader or the Cloud. From now on, the tracking device will send 2-hourly (configurable) status updates and reports of optional sensors attached to the

cargo, again according to UTC time. In addition, it will immediately report about any critical event that occurs.

Finally, there is an optional mode called Secured Vessel Mode (SVE mode). This mode is initiated when the tracking device receives a trigger from a fixed router, installed on the loading quay, or from the Cloud (when the device connects to the Cloud). After receiving the trigger, the tracking device can go to energy-saving mode, where the IEEE-802.15.4 radio is disabled, an internal clock is scheduled, and no status updates are sent until the device wakes up again. This mode can be used when the container is loaded in a vessel. An energy-efficient backup mechanism ensures that the tracking device will eventually wake up in case something goes wrong.

The different modes can meet the identified monitoring requirements, while at the same time taking care of the energy consumption: most energy is consumed when the container is sealed, less energy is consumed when the container is empty, and least energy can be consumed when the container is on a vessel and cannot be accessed until arrival of the vessel. Also note that at all times, when no UMTS/GPRS or GPS communication is required, the respective modules are turned off.

4.3 Communication overview

Based on this general architecture, we have derived a number of scenarios for which communication solutions have been designed, as illustrated in part b of Fig. 11.

The transition between the different modes (REST, STP, and SVE) is depicted at the top of part b in Fig. 11. When a container is sealed for transport, it must go from REST to STP mode. This is done by an operator that sends a trigger using a MoCo reader (type 5). Optionally, in STP mode, intra-container monitoring can take place (type 4, dark red). The sensors inside containers, attached to the cargo, either buffer the (delta) values until the tracking device requests the sensor data (just before report transmission) or transmit the sensed data immediately when a critical threshold was exceeded (type 3). When a container is put on a vessel, the device goes to SVE mode after receiving a trigger from a MoCo router (type 6). This results in the device sleeping (type 3, light red) until it is woken up, triggering it to go back to STP mode.

As mentioned, devices report X -hourly in REST and STP mode according to a specified UTC time. Therefore, we can either have a group of devices not transmitting data (type 1, dark green) or a group of devices transmitting data (type 2, light green). When in type 2, there is first a transmission setup period where devices get the opportunity to select a traffic device as gateway, collect the reports at this gateway, and, eventually, forward the collected reports to the cloud. This is illustrated at the bottom of part b in Fig. 11.

Making a distinction between a non-transmitting period and a transmitting period enables us to take additional energy-saving precautions in the non-transmitting period. Further, for all communication with the outside world, a communication hierarchy has been identified, specifying that IEEE-802.15.4 technology must be preferred over UMTS/GPRS technology and, when using UMTS/GPRS technology, this must be done in such a way to reduce roaming cost and energy consumption.

4.4 Tailored sensor communication solutions

Based on the previous specifications, a communication architecture and communication protocols have been designed for the tracking device (and router/reader). The architecture and protocols aim to implement the specs in an energy-efficient way. To this end, several novel and innovative features have been incorporated in the architecture, mostly related to the inter-container communication over the IEEE 802.15.4 radio. We will only highlight the most important components.

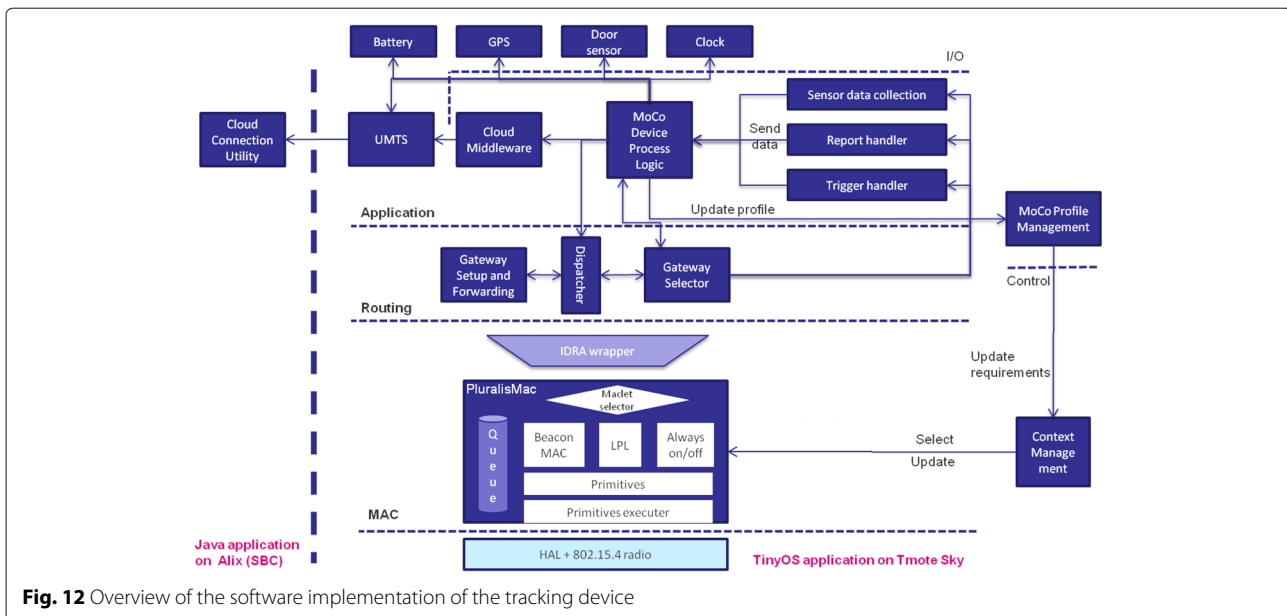
In Fig. 12, this high-level communication architecture of the MoCo device is presented. It consists of several building blocks dealing with the MoCo process logic, UMTS, and 802.15.4 communication and all required networking functionality. The components are implemented on

a board running an TinyOS. This board has a 802.15.4 radio interface on top of which a hardware abstraction layer is provided in order to make use of the underlying hardware functionality. Similarly, a UMTS/GPRS module and corresponding API provide direct connectivity to the outside world. In the following paragraphs, the different building blocks, together with their role in the system, will be briefly explained and some specific innovative features will be emphasized.

The Device Process Logic component is the heart of the application running on the embedded device. It implements and executes all process logic such as the mode changes of the tracking device, the behavior within a specific mode, and changes in status. As such, this component will interact with many other modules. All specific information such as report intervals, duty cycles, and timing values are captured in a device profile. This profile can be updated, upon which other components can be informed and reconfigured.

Just above the HAL and IEEE-802.15.4 radio, we have PluralisMAC [15]. From a MAC layer point of view (IEEE 802.15.4 MAC), the specifications reveal that in the non-transmitting period (REST or STP mode), the radio should sleep as much as possible, while remaining capable of receiving triggers or sending a critical event. In the transmitting period (REST or STP mode), data packets should be sent as efficiently and reliably as possible. In SVE mode, it should be disabled completely. This results in completely different requirements on the IEEE-802.15.4 MAC layer and thus requires different MAC strategies and scheduling. Current MAC designs are monolithic, and PluralisMAC provides an innovative solution. It supports different MAC strategies on a single device through reuse and intelligent combination of primitives (e.g., data, ACK, BEACON, RST) into maclets. For example, low-power listening, beacon MAC, and always off are supported, which are tailored to the non-transmitting period, transmitting period, and SVE mode respectively.

At the routing/networking layer, we have the Dispatcher, Gateway Selector and Gateway Setup, and Forwarding components. The role of the Dispatcher is simple. It delivers incoming packets from the MAC layer to the appropriate module. In the other direction, it handles outgoing packets that need to be transmitted over the IEEE 802.15.4 radio interface. The Gateway Selector is an important component that implements the gateway selection protocol. Through this component, it is possible to select a gateway between neighboring tracking devices. Figure 13 illustrates the gateway selection algorithm by means of a simplified flowchart. The novel low-overhead algorithm allows selecting the best (= device with most remaining energy) gateway in a 1-hop neighborhood based on a gateway willingness metric. Apart from the achieved reduction in energy consumption through UMTS/GPRS



sharing, the protocol strives to achieve fairness in energy consumption. Basically, it constitutes of two stages: (1) check if another node is announcing itself as a gateway, and (2) if no other nodes are announcing, start announcing yourself as a gateway. The mechanism works in such a way that the node with the highest energy starts announcing first. If the clock drift is acceptable, the energy consumed during this procedure is minimal. In the first stage, only periodical checks for medium activity (LPL) are required and, optionally, periodically send beacons to other nodes in the second stage. The gateway selection is limited to the 1-hop neighborhood in order to make the algorithm robust. We preferred using an algorithm that is simple instead of an algorithm that searches the most optimal gateway. Searching such an optimal gateway would consume a lot of energy, and introducing multihop communication would make the algorithm more complex and would introduce more link errors.

Once a gateway has been selected, the Gateway Setup and Forwarding component in the involved MoCo devices will establish a correct routing path to the selected gateway (which is either a router or another tracking device). When a device use another device (or router) as gateway, the selected gateway will receive the reports from that devices. As said, the Dispatcher delivers incoming packets to the appropriate module. For example, incoming triggers from a reader of router are delivered to the Trigger Handler component. This component will process the trigger (packet processing, acknowledging) and will then inform the Device Process Logic about the type of trigger that needs to be executed. It is also possible that data from intra-container sensors is received. This data is delivered to the Sensor Data Collection component, which stores it

until the next transmission moment where the data can be delivered to the Cloud. When a device use another device (or router) as gateway, the selected gateway will receive the reports from that devices. These reports are delivered to the Report Handler, which stores them until the next transmission moment where the data can be delivered to the Cloud by the Cloud Middleware component. This component is responsible for the preparation of all data that needs to be transmitted to the Cloud over the UMTS/GPRS interface of that device. This data not only includes its own report but can also include collected sensor data or reports from neighboring devices that have selected this device as their gateway.

The architecture also contains several I/O modules that provide interfaces to mostly external hardware components such as the GPS module, GSM/UMTS module, door sensor, and battery. The nodes are synchronized in two ways: (a) directly when retrieving the GPS location and (b) indirectly via the 802.15.4 inter-container network (in this case, the GPS time of the 802.15.4 gateway is added in the acknowledgement of the reports send through the gateway). The gateway selection algorithm use guard times that take into account the clock drift on the node.

5 Process validation

Once the system architecture and the process had been defined, it was necessary to reflect whether the envisaged system would be able to achieve the desired requirements. Especially, the energy consumption of the tracking device is a major concern. Therefore, we have calculated worst-case estimations for the device lifetime based on the number of transmissions that the designed process would yield for different configurations. The estimations are based on

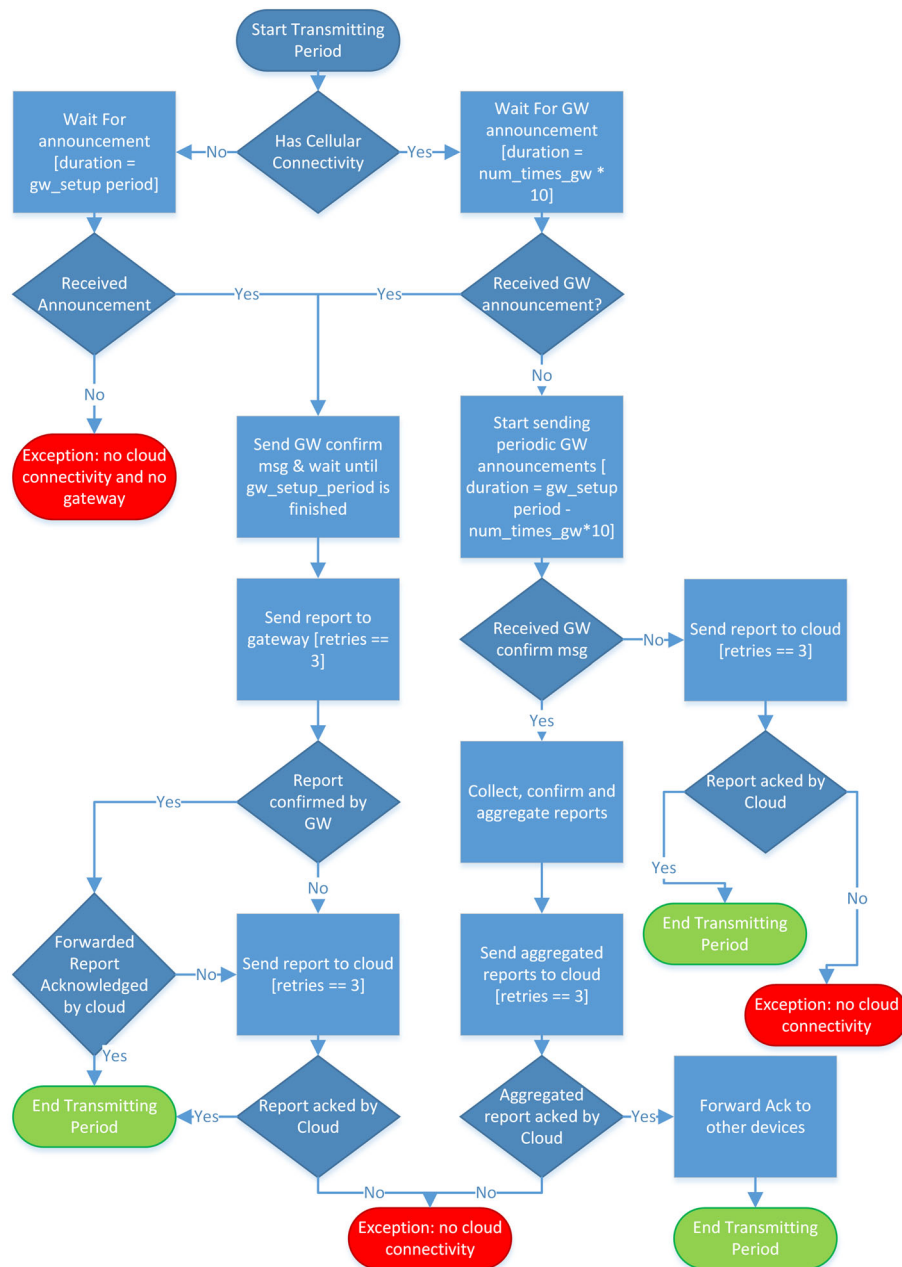


Fig. 13 Simplified flowchart illustrating the gateway selection algorithm

information (power consumption, timing) obtained from datasheets of the u-blox LISA-U1 GPRS/UMTS, u-blox NEO-6 GPS, and TI-CC2420 modules. The relevant input parameters for power consumption for the UMTS, GPS, and 802.15.4 radio modules and microcontroller are listed in Table 4. Note that always, worst-case values are used and that there are more efficient modules on the market today. Moreover, also for the timings (listed in Table 5), conservative values are used for UMTS connection setup and transmissions, and GPS position acquisition (e.g., cold

start). The other timing input variables are based on the envisaged process and can also be further optimized.

It is important to know how the tracking device will behave in the worst-case scenario: no inter-container communication, thus no UMTS/GPRS sharing, and never in the more energy-efficient SVE mode. In this scenario, the tracking device always uses its own UMTS/GPRS and GPS module every x -hour. The IEEE-802.15.4 radio will be active, with a higher duty cycle during transmitting periods than during non-transmitting periods. This

Table 4 Power requirements for the process validation calculations

Power requirements (watt)		
UMTS module	Standby	0.000342
	Active	2.736
GPS module	Standby	0.000066
	Active	0.141
802.15.4 radio	Standby	0.000027
	Active	0.075
MCU	Standby	0.0001635
	Active	0.0054
	LPM	0.0000153

worst-case scenario reflects the situation during an initial rollout, i.e., there are almost never neighboring tracking devices, and in the absence of SVE mode. The results shown in Fig. 14 therefore illustrate worst-case behavior and can be enhanced significantly when tracking devices can share UMTS/GPRS connections over the 802.15.4 network as demonstrated in the next paragraphs.

Part a of Fig. 14 shows the daily energy consumption for the UMTS/GPRS module, GPS module, and IEEE-802.15.4 radio of a tracking device for different reporting intervals. The UMTS/GPRS energy consumption for short intervals is very high compared to IEEE-802.15.4. This is no surprise considering that a typical UMTS module consumes 2.736 W compared to 0.075 W for the

CC2420 radio, both in active mode. The benefit of inter-container communication becomes clear when considering two tracking devices sharing their UMTS/GPRS/GPS connection every 2 h. Now, each device only has to make a connection every 4 h. Since the energy consumption of IEEE-802.15.4 is nearly constant, this resembles the situation where one tracking device reports every 4 h, leading to a considerable reduction in energy consumption.

Part b of Fig. 14 shows the expected battery lifetime for different report intervals and for a different number of days in Secure Transport Mode (STP). The STP-mode is most energy-consuming because the reporting interval will be highest and also the door sensor needs to be monitored. We can conclude that in case a battery lifetime of 1–3 years is desired, either the report interval or the number of days in STP must be limited. Since the number of days in STP is dependent on the usage of the container, the only possibility is to let the report interval vary based on the value of the cargo and the days already spent in STP. Considering the same example as above, the number of days in STP could almost double when two tracking devices share their UMTS/GPRS/GPS.

Combining both conclusions, we can assume the following:

Assumption 6 *Tracking devices in the envisaged system can have a battery lifetime of 1–3 years if either enough tracking devices share one UMTS/GPRS/GPS connection or if the report interval can be varied.*

6 Test lab evaluation

After confirming the feasibility of the envisaged process, the next step was to evaluate the novel sensor protocols in a controlled environment. This is done using our wireless test lab deployed in an office environment [16]. There are two aspects that we wanted to evaluate: (a) the behavior of the gateway selection algorithm and (b) the different MAC logics, implemented using pluralisMAC.

6.1 Gateway selection algorithm

Because an office environment has different wireless characteristics than a stacked container environment, results obtained on the test lab would not be representative for testing the behavior of the gateway selection algorithm. For this reason, a new tool was developed that is able to emulate different wireless environments and network configurations. This tool works on top of the existing test lab infrastructure.

Part a of Fig. 15 shows both the test lab components and the wireless link emulator (WLE) overlay. The test lab consist of four major components: a central server running Linux, an Alix 3d3 embedded single board computer (SBC) running Linux, an Environment Emulator (EnvEmu) running TinyOS, and a Tmote Sky sensor node

Table 5 Timing input for the process validation calculations

Timing (unit)		
UMTS connection	15	s
GPS position fix	90	s
802.15.4 duty cycle (active)	0.01	%
802.15.4 duty cycle (standby)	0.1	%
STP/REST transmitting period	300	s
STP/REST gateway search	60	s
STP/REST gateway setup	60	s
STP/REST report collection	120	s
STP/REST report ack	60	s
STP report interval min	3600	s
STP report interval max	64,800	s
REST report interval	86,400	s
SVE check interval	43,200	s
SVE UMTS search	300	s
MCU active	200	s
MCU standby	100	s
MCU LPM	X – 300	s

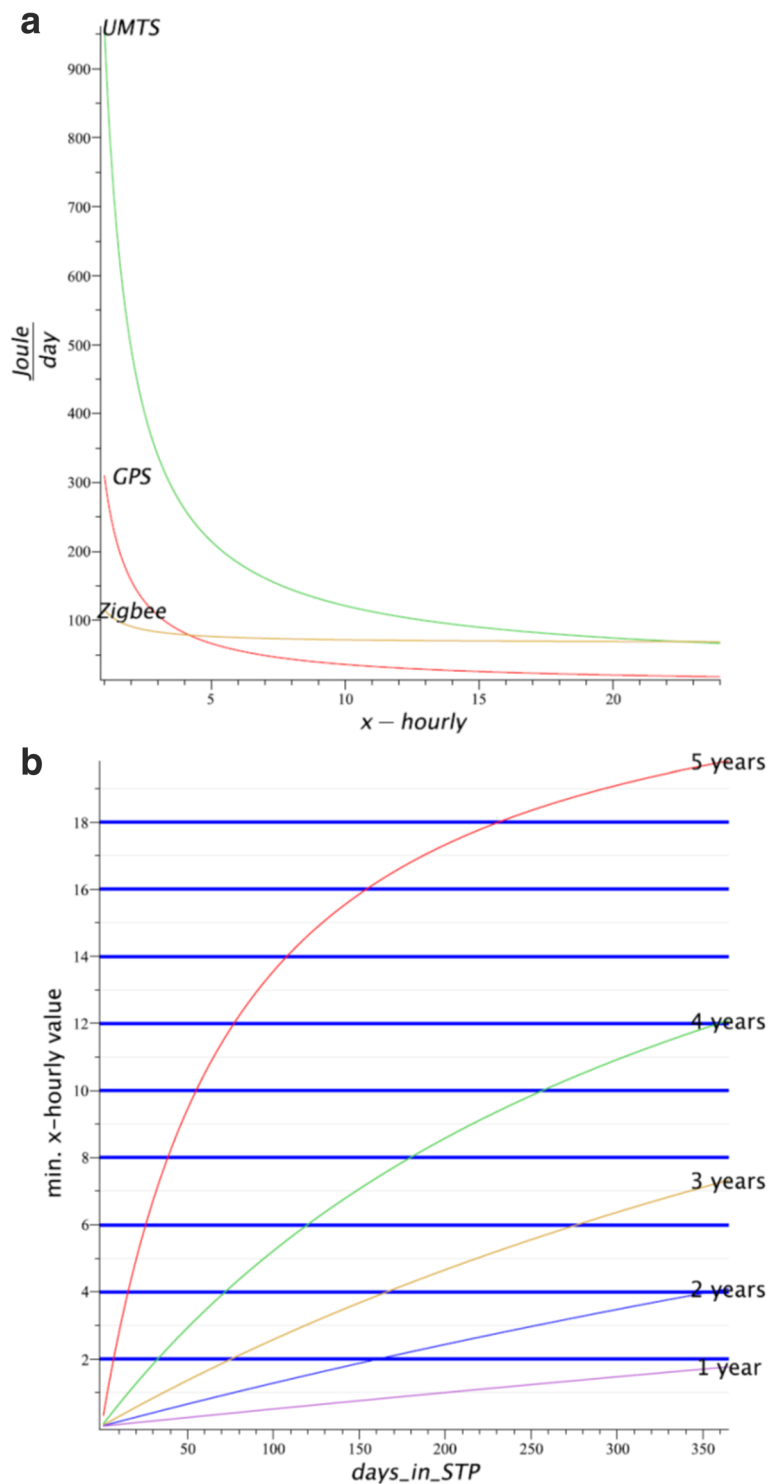


Fig. 14 Validation of the process

running TinyOS (device under test, DUT). The *central server* controls the experiments (on SBC or DUT), configures the SBC and the EnvEmu, gathers and stores results, and manages users in our test lab. Using

WLE functionality, it provides libraries to control the network topology, to apply the path loss models, and to forward packets to all receivers based on the topology and model.

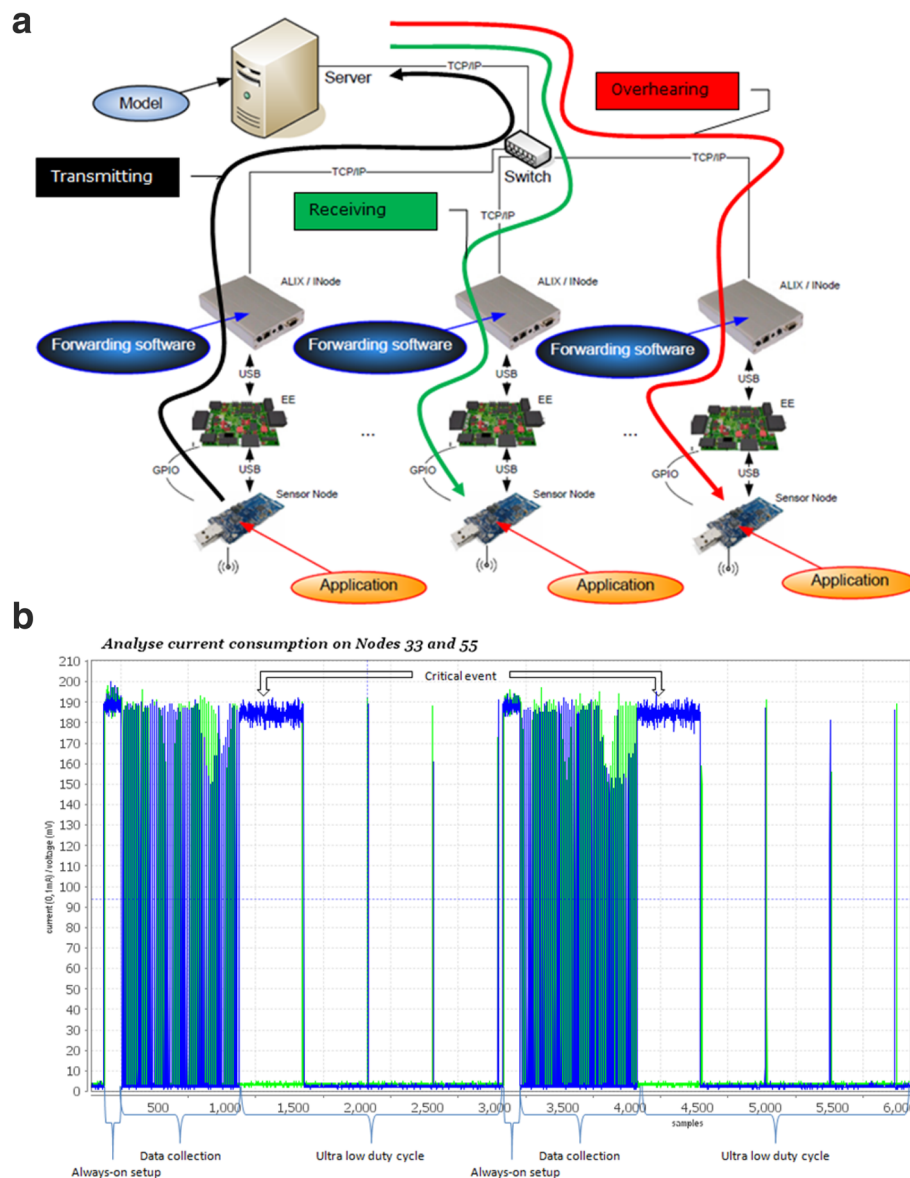


Fig. 15 Test lab evaluation

The *SBC* runs administrative software to forward serial control messages to EnvEmu and to program the DUT. It can also be used to execute user-defined software. Extra-WLE functionality was added to forward packets received from the DUTs to the central server (transmitting arrow) and to forward packets from the central server to the DUTs (receiving or overhearing arrow). For the evaluation of the container monitoring solution, it also emulates a UMNIS/GPRS modem and a GPS locator.

The *Environment Emulator* enables sensor and battery emulation for the DUT. It can also be used to monitor GPIO pins and measure energy consumption of the DUT.

On the WLE overlay, it provides the possibility to emulate an SFD pin for synchronization purposes.

The *Imote Sky* or *DUT* executes user defined sensor software both for the regular test lab as for the WLE. In a WLE experiment, it forwards radio messages over serial to the SBC and receives radio messages over serial from the SBC.

Note that the topology and the wireless model used on the central server are defined in Section 3 and can be altered with little effort. This enables us to emulate more than just one wireless model.

The *gateway selection algorithm* was tested using the wireless link emulator in different scenario's in order to

identify limitations. For this purpose, we increased the number of containers in every iteration, always adding one container at the border of the stack. We also considered two positions for the gateway, one in the middle of the stack (best-case) and one at the border of the stack (worst-case). We discovered that the gateway selection starts to fail when working on the boundary of the wireless range. In the worst-case gateway position with six containers, we observed that only 10% of the transmission cycles¹ was successful. Since this is a common scenario in real-life, we have added an RSSI filter, avoiding that tracking devices select a gateway to which they have a too low-quality link. After applying the RSSI filter, the network size is limited to five containers on each side of the gateway. We still noticed that for the device on the far edge of the stack 20% of the transmission cycles failed. Using its built-in fallback mechanism, the device used its own UMTS/GPRS module to report to the cloud. All other devices could successfully complete nearly all transmission cycles.

6.2 MAC logic in PluralisMAC

One of the most important component to evaluate, apart from verifying the correct execution of the process logic, was the newly designed MAC layer, PluralisMAC. *PluralisMac* was evaluated using the energy measurement features of the environment emulator. Part b of Fig. 15 shows the current consumption on two nodes (one sample every 125 μ s). Node 55 (blue line) was configured as gateway, and node 33 (green line) was one of the nodes using this gateway. The different stages of the transmitting-non-transmitting cycle can be clearly identified. The transmitting period starts with a small always-on gateway selection and setup stage. This is followed by the data collection stage where reports and sensor data are collected. Now, PluralisMac is configured to use a beacon MAC. Finally, during the non-transmitting period, LPL is used with an ultra-low duty cycle. In the beginning of this period, a critical event is transmitted from node 33 to node 55 (green line). A more thorough evaluation can be found in [15].

6.3 Problems

The evaluation on the test bed allowed us to verify the behavior of most key components and already revealed some issues that could be dealt with at this stage of the design. However, there were also some aspects we could not evaluate on the test lab due to practical reasons or shortcomings of the Wireless Link Emulator. First of all, we could not perform a proper evaluation of PluralisMac in a virtual container environment. This is due to the fact that emulated packet transmissions have a higher delay than genuine radio packet transmissions, which makes it impossible to evaluate MAC protocols. The additional delay is introduced by the longer path between

senders and receivers. Moreover, the broadcast nature of wireless transmission is emulated using unicast messages to all receivers (both destined receivers as overhearing receivers). The average delay can be formulated as follows:

$$D_{\text{packet}} = D_{\text{serial}} + D_{\text{networkup}} + D_{\text{processing}} + \frac{\sum_{i=1}^N D_{\text{networkdown}}}{N} + D_{\text{serial}} \quad (3)$$

Secondly, there were also issues while doing energy measurements with the EnvEmu when using the Wireless Link Emulator. Because serial communication is required, the measured energy is not representative. Therefore, we performed the energy measurements in the actual office environment of our test lab. Finally, since the test lab is not equipped with a UMTS/GPRS modem and GPS locator, these had to be emulated by the embedded SBC.

7 Field trial validation

After successful evaluation in our test lab, we were finally ready to build the actual prototype for use in an initial field trial. This included the integration of a UMTS/GPRS module, the GPS locator, and the door sensor. This was the final step of the project and should lead to a prototype that is ready for further industrialization. If the field trial turned out to be successful, the prototype could be further enhanced by the industrial partners and put to a long-term test. This was however out of the scope of the project.

7.1 Experiment setup and configuration

The field trial was performed in a container stacking similar to the test lab setup. First, a number of prototype devices were built, consisting of four major components: a RM-090 sensor board [17], an airlink fastrack xtend GPS-enabled GPRS modem [18], a battery pack, and a door sensor. The prototype devices (part b of Fig. 16) were installed on the front left side of the container (when looking to the door). The antennas are mounted on the ventilation cap of the container. The actual setup is shown in part c) of Fig. 16. Again, we considered two positions of the gateway, a best-case position and a worst-case position (devices 1 and 4 respectively on the figure).

The devices were configured to send both regular reports and sensor data (RSSI and LQI samples). This allows us to compare the link quality during the validation with the link quality observed when the path loss model was measured. All tracking devices were battery powered except the gateway². This device was attached to a laptop computer in order to monitor the behavior of the software and to collect the RSSI and LQI samples. The regular reports were sent via GPRS to a web server that displays container status information. Part a of Fig. 16 shows a screenshot of this during the field trial.



Fig. 16 Field trial

7.2 Results

During the field trial, we checked if the gateway selection algorithm worked as expected. If so, then the assumptions made in Section 3 were also valid. We observed that nearly all transmission cycles were successful for every device for the best-case gateway position. We also observed that device 4 had the lowest RSSI values (avg. -90 dB). This can be explained by considering that device 4 is on the edge of the stack and it is not surrounded with other containers. This limits the opportunities for multipath propagation. For this reason, we chose device 4 as the worst-case gateway position.

For the worst-case gateway position, we could clearly see that device 5, located at the other edge of the container stack, has high error rates. More than 40% of the transmission cycles fail, which is a bit more than expected. The average RSSI for device 5 is also very low (-94 dB). During this scenario, we could also observe that small packets have more probability of success than big packets. Report messages (42 bytes) originating from device 5 were

successfully transmitted in 90% of the cases. The RSSI data messages (100 bytes) only had a success rate of 40%. Note that if either one of both fails, the entire transmission cycle was flagged as failed.

Since it was possible to create an inter-container network spanning more than four containers in a row configuration, we can conclude that assumptions 1 and 2 hold. Assumption 3 does not hold entirely because we observed 20% more failures than during the test lab evaluation. This can be partially explained by the difference in the container stacking. The orientation of the containers was different, leading to worse multipath propagation during the validation experiment. Moreover, also, the distance between the far edges was larger than during the experiment in the test lab. It is however a situation that will frequently occur in real life.

7.3 Problems

Due to practical limitations, the containers could not be placed in the same orientation. Moreover, we could only

use one 40-ft container and four 20-ft containers. We also noticed that the best-case scenario differs. This is because the orientation and the vicinity of other containers (that can reflect the signals) have a big impact on the multipath propagation. We also observed that packet length has a high impact on the probability of success when suffering from bad multipath propagation. Shorter packets have a higher probability of success. The model used in the wireless link emulator should be enhanced by incorporating these observations.

We could not validate assumption 4 (Section 5). A long-term deployment is necessary to check if we are able to achieve the desired battery lifetime. This is a very important aspect of the system but was not possible in the scope of this project. Long-term experiments are scheduled for the new prototype that is currently under development and being enhanced based on the outcome of the field trial validation.

8 Conclusions

In this work, we have presented a prototype for container monitoring. We described the methodology that enabled us to create a tailored solution for container monitoring with specific requirements under challenging conditions.

We started with a feasibility study to establish if we could realize our goals. This study gave us enough insights and enabled us to use certain assumptions we could use in the design phase to create innovative solutions. For instance, the ability of switching MAC strategies on the fly and to select the best gateway in terms of energy offers important pieces of research. After the design phase, we first validated if we could obtain the desired battery lifetime by modeling the system behavior. We could clearly see that the system benefits from the proposed inter-container communication but we also identified a worst-case device lifetime. Then, we could start develop and test on our test lab using a wireless link emulation tool to evaluate the software under realistic conditions. During this stage, we could optimize our solution and tackle specific issues with the developed software. We identified the need for RSSI filtering in order to create a reliable, robust, and low energy network. We also validated if the proposed prototype could work in an operational environment during a field trial. Although the results differed a bit from the test lab evaluation, we were still able to use our solution without any modifications. This convinced interested parties to invest in the proposed system and to create an industry grade product.

We believe that sensor technologies will make their entrance in more and more areas. Therefore, it will be increasingly important to evaluate solutions in the correct context and to assess feasibility of concepts and ideas at an early stage enabling rapid prototyping. We presented a specific method that fitted our needs but believe that

in the future, methods and tools should be standardized. This should start at an early stage still considered to be research rather than product development.

Endnotes

¹ In a transmission cycle, a group of tracking devices is transmitting. We only consider the period where the IEEE-802.15.4 radio is active and the tracking devices execute the gateway selection algorithm, set up an inter-container network, and collect reports.

² Because we wanted to create a container stacking similar to the test lab setup, we predetermined the gateway by giving each device a fixed gateway willingness.

Acknowledgements

The iMinds MoCo project is co-funded by iMinds, a research institute founded by the Flemish Government. Companies and organizations involved in the project are Multicap and Rmoni, with project support of IWT (Agency for Innovation by Science and Technology in Flanders).

Authors' contributions

PR conceived, designed, and implemented the inter-container communication feasibility study, the system architecture, and communication stack and performed and analyzed the test lab evaluation and the Field Trial Validation. PR, JH, and EDP wrote the paper. IM supervised the work. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 4 August 2016 Accepted: 5 January 2018

Published online: 15 January 2018

References

1. iMinds MoCo project website. <https://ibbt.be/en/projects/moco>. Accessed 10 Jan 2018
2. On the use of RFID in the management of reusable containers in closed-loop supply chains under stochastic container return quantities. *Transp. Res. Part E Logist. Transp. Rev.* **64**, 12–27 (2014)
3. B Lannoo, B Naudts, E Van Hauwaert, P Ruckebusch, J Hoebeke, I Moerman, in *Proceedings of WCCTRS-SIG2*. Techno-economic evaluation of a cost-efficient standard container monitoring system (Elsevier, Antwerp, 2012)
4. S Mahlknecht, SA Madani, in *2007 5th IEEE International Conference on Industrial Informatics*. On architecture of low power wireless sensor networks for container tracking and monitoring applications, vol. 1 (IEEE, Vienna, 2007), pp. 353–358. <http://doi.acm.org/10.1109/INDIN.2007.4384782>
5. S Schaefer, in *Companion to the 21st ACM SIGPLAN Symposium on Object-oriented Programming Systems, Languages, and Applications*. Secure trade lane: a sensor network solution for more predictable and more secure container shipments (ACM SIGPLAN, Portland, 2006)
6. W Lang, R Jedermann, D Mrugala, A Jabbari, B Krieg-Bru? andckner, K Schill, The intelligent container: a cognitive sensor network for transport management. *IEEE Sensors J.* **11**(3), 688–698 (2011). <http://doi.acm.org/10.1109/JSEN.2010.2060480>
7. K Delaney, D Rogoz, F O'Reilly, in *Ambient Intelligence with Microsystems*. Dedicated networking solutions for a container tracking system (Springer US, Boston, MA, 2009), pp. 387–408. https://doi.org/10.1007/978-0-387-46264-6_18
8. R Sharma, Y Lee, BM Kim, YJ Kim, YG Heo, HJ Jeon, KH Kim, SR Lee, in *2013 International Conference on ICT Convergence (ICTC)*. Auto location and

- security alert embedded with container identification for real time applications (IEEE, Jeju, 2013), pp. 365–370
9. EK Lee, SP Choi, YS Moon, TH Kim, BH Lee, CS Kim, JJ Kim, HR Choi, in *16th International Conference on Advanced Communication Technology*. A study on the performance evaluation of container tracking device based on m2m (IEEE, Seoul, 2014), pp. 67–72
 10. K Langendoen, A Baggio, O Visser, in *Proceedings of the 20th International Conference on Parallel and Distributed Processing*. Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture (IEEE, Rhodes Island, 2006)
 11. M Ceriotti, M Corra, L D'Orazio, R Doriguzzi, D Facchin, ST Guna, GP Jesi, RL Cigno, L Mottola, AL Murphy, M Pescalli, GP Picco, D Pregolato, C Torgele, in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference On*. Is there light at the ends of the tunnel? wireless sensor networks for adaptive lighting in road tunnels (IEEE, Chicago, 2011)
 12. L Mottola, GP Picco, M Ceriotti, c Gună, AL Murphy, Not all wireless sensor networks are created equal: a comparative study on tunnels. *ACM Trans. Sen. Netw.* **7**(15), 15:1–15:33 (2010). <http://doi.acm.org/10.1145/1824766.1824771>. <https://doi.org/10.1145/1824766.1824771>
 13. G Barrenetxea, F Ingelrest, G Schaefer, M Vetterli, in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. The hitchhiker's guide to successful wireless sensor network deployments (ACM, Raleigh, 2008)
 14. G Barrenetxea, F Ingelrest, G Schaefer, M Vetterli, O Couach, M Parlange, in *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference On*. Sensorscope: out-of-the-box environmental monitoring (ACM, St. Louis, 2008)
 15. P De Mil, P Ruckebusch, J Hoebeke, I Moerman, P Demeester, Pluralismac: a generic multi-mac framework for heterogeneous, multiservice wireless networks, applied to smart containers. *EURASIP J. Wirel. Commun. Netw.* **1**, 166 (2012). <http://doi.org/10.1186/1687-1499-2012-166>. <https://doi.org/10.1186/1687-1499-2012-166>
 16. S Bouckaert, W Vandenbergh, B Jooris, I Moerman, P Demeester, in *6th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks Communities*. The w-ilab.t testbed (Springer, Berlin, 2010)
 17. Product website of the RM090 sensor board (2012). <http://www.rmoni.com/en/products/hardware/rm090>. Accessed 10 Jan 2018
 18. Product website of the GPRS modem. *** (2012). <http://www.sierrawireless.com/>. Accessed 10 Jan 2018

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com